Requirements Document

Lab 3 – Alarm Clock

# Overview

## Objectives

The objectives of this project are to design, build and test an alarm clock using the TM4C123 microcontroller and various circuit components. Some of the main goals include building and debugging modular software, processing user input, and to remove critical sections within our software. We will also learn how to generate sound using a square wave on a speaker component.

## Process

The alarm clock project will be developed using the TM4C123 board, and the on-board switches will be used to handle user input. The system and all of its components will be built on a solderless breadboard and run on USB power. The on-board LED will be used as a heartbeat for the system. The speaker will be external and connected to the breadboard. There will be five hardware/software modules: switch/keypad input, time management, LCD graphics, heartbeat, and sound output. We will design and test each module independently from the other modules. After each module is tested, the system will be built and tested as a whole.

## Roles and Responsibilities

The TA is our client for this system, and we (the students) are the engineers that were tasked to build the alarm clock. The purpose of this project is to build the system but also to learn how it works, and so it is the responsibility of the students to understand all parts of the system, even if they didn’t work on that part. Because the system should be modular, we will split up the work between the 2 partners on the team, but we will both understand every component of the system for the demo.

## Interactions with Existing Systems

Existing systems that our project will use include a TM4C123 microcontroller board, a solderless breadboard, and a ST7735 color LCD screen. The components will be connected together using the breadboard, and the microcontroller will be powered with a USB cable which then powers the entire system.

## Terminology

|  |  |
| --- | --- |
| Power budget | Maximum power that the power supply can drive. This limits the number of components that you can connect to a supply |
| Device driver | Interface between a device and a software module. An example would be the functions defined in ST7735.c |
| Critical section | Multiple threads access shared memory locations/variables. A higher priority thread has to interrupt another thread and write to a variable that the lower priority thread was reading. |
| Latency | The time between a service request and the process actually being serviced |
| Time jitter | The difference in the maximum latency and the minimum latency that occurs. Some systems have a strict minimum time jitter requirement. |
| Modular programming | Break up your software into modules with distinct tasks. Each of the modules should be kept separate so that you can easily debug or change specific features of your system. |

## Security

Software from Tivaware will not be used in this project. Instead, the software will interact with the board at a low level and directly write to the appropriate registers. Code written for this project can’t be shared with any other EE445L students (past, present, and future). It is the responsibility of both partners to ensure that their code does not fall into the hands of other students.

# Function Description

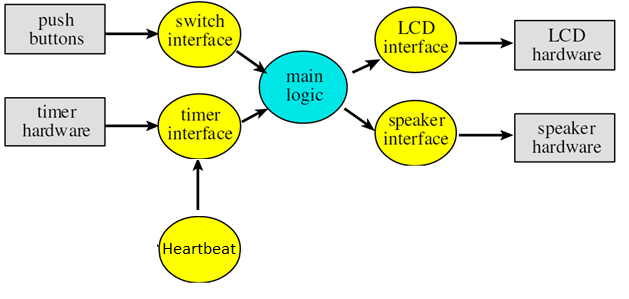
This section will detail the specific functionality of the system by listing the expected functions and describing how the components of the system are connected.

## Functionality

The alarm clock system will be able to perform the following functions:

1. Display hours and minutes in both graphical and numeric forms on the LCD. The graphical output will include the 12 numbers around a circle, the hour hand, and the minute hand. The numerical output will be displayed in the corner on the same screen.
2. Set the current time by holding down SW1 (on-board) for 2 seconds.
3. Set the alarm time by holding down SW2 (on-board) for 2 seconds
4. Set the alarm type by holding down both SW1 and SW2 (on-board) for 2 seconds. SW1 and SW2 are then used to cycle through the possible alarm types. To select an alarm, the operator will press both SW1 and SW2.
5. Sound the alarm when the alarm time is hit. The alarm type that is played should be the one that the user chose
6. Stop the alarm by pressing any on-board button while the alarm is sounding.
7. LED heartbeat to show that the system is running

## Software Diagram



## Scope

The lab is divided up into 3 parts: preparation, demo, and lab report. In the preparation, we are expected to have software written for each module as well as a main program to test the alarm. For the demo, we will show the TA our system with the alarm-selector as an extra feature. We will then answer any questions that the TA may have for our team. For the lab report, we will include this requirements document, a software diagram, hardware specifications, and answers to the questions shown in the lab manual.

## Prototypes

A prototype system running on the TM4C123 board, ST7735 color LCD, and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

## Performance

The system will be judged by three qualitative measures:

1. The software modules must be easy to understand, well-organized, and modular.
2. The clock display should be aesthetically pleasing and accurately display the time.
3. The operation of setting the time and alarm should simple and intuitive.

The system does not have any critical sections. Backward jumps in the ISR should be avoided if possible. User input is handled in the main program, but updating the LCD is done in an ISR. This isn’t ideal because it increases the time for the ISR to finish, but it solved our problem of critical sections. Having the ISR that updates the time also update the display ensures that no 2 threads are trying to access this shared time variable. The average current on the 5V power will be measured with and without the alarm sounding. This will show how much power the alarm uses while it is sounding

## Usability

User input is handled by using the 2 on-board switched on the TM4C123 microcontroller. The following list shows all of the input options for the user:

1. Set time – Hold down SW1 for 2 seconds. SW1 and SW2 are then used to increment the hours and minutes respectively
2. Set alarm – Hold down SW2 for 2 seconds. SW1 and SW2 are then used to increment the hours and minutes respectively
3. Change the alarm type – Hold down both SW1 and SW2 for 2 seconds. SW1 and SW2 are then used to cycle through the list of alarm types
4. Turn on/off alarm – press any button while the alarm is sounding

The user will be able to set the time (hours, minutes) and be able to set the alarm (hour, minute). There are no seconds displayed for the clock, and so the user can not set that value. When the time is set, the seconds are automatically set to 0. The digital time is displayed in military time with a rollover at 23:59.

The LCD display shows the time using a graphical display typical of a standard on the wall clock. The 12 numbers, the minute hand, and the hour hand are large and easy to see. The clock will also display the digital time in the top-left corner. Instructions on how to use the clock are also displayed in that corner.

## Safety

The alarm sound will be VERY quiet in order to respect other people in the room during testing. A resistor connected to the base of the transistor is used to control the loudness of the speaker. Connecting or disconnecting wires on the protoboard while power is applied may damage the board.

# Deliverables

## Reports

A lab report is due on 09/23. The lab report will contain this requirements document, software and hardware designs, and answers to a few questions defined in the lab manual.

## Audits

The preparation for the lab is due on 09/15 and will be graded by the TA.

## Outcomes

There are three deliverables: preparation, demonstration, and report.